It's Not Easy Being Green

DiPS CodeJam 24

Prompt

'It's not easy being green,' was the frequent refrain of Kermit the frog.
'It seems you blend in With so many other ordinary things,'
he says. Let's find him some friends, shall we? Each input is a string of length n made up of numbers and uppercase letters. Your task is to find the number of 6-character substrings which, when read as a hex colour code, are shades of green.

Input Format

- The first line of the input contains an integer n, denoting the number of conferences.
- The next n lines of the input each contain the start and end dates of a conference, in the format $(\text{start}, \text{end})$ in a space-separated list.

Output Format

The first and only line of your output must contain a single integer m, denoting the maximum number of conferences they can attend.

Constraints

- $10^4 \leq n \leq 10^6$
- Assume that the conferences are already sorted based on end dates.
- Assume 365 days in a year.
- Assume that dates are given in the form of a single integer n denoting the n^{th} day of the year.

Sample Input/Output

Input	Output

Solution

This is an example of the *Activity Selection Problem.*

Simplifying the Problem

Assume there exist n conferences with each of them being represented by a start time s_i and finish time f_i. Two conferences i and j are said to be non-conflicting if $s_i \geq f_j$ or $s_j \geq f_i$. The activity selection problem consists in finding the maximal solution set (S) of non-conflicting conferences. Here, using a greedy algorithm to find the solution will always result in an optimal solution.

Solving the Problem

- Let us create an empty array $a[]$.
- Now we can start adding conferences to this array.
- Since this is a greedy algorithm, the first conference is always selected.
- Now we loop through the rest of the conferences. For each conference:
  - If this conference has a start date that is greater than or equal to the finish date of the previously selected conference, then append it to $a[]$.
- Finally, we print the length of $a[]$, denoting the number of conferences.

Sample Program

```
# n --> Total number of conferences
# s[]--> An array that contains start time of all conferences
# f[] --> An array that contains finish time of all conferences

n = int(input())

s = []
f = []

for i in range(n):
     inputArr = list(map(int, input().split()))
     s.append(inputArr[0])
     f.append(inputArr[1])

conferences = []

# The first activity is always selected
i = 0
conferences.append(i)

# Consider rest of the conferences
for j in range(n):

     # If this activity has start time greater than
     # or equal to the finish time of previously
     # selected activity, then select it
     if s[j] >= f[i]:
          conferences.append(j)
          i = j

print(len(conferences))
```